4 Logistic regression : classify emails as spam or not using the spam dataset

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.learn_model import LogisticRegression
from sklearn.metrics import classification_report,
confusion_matrix, accuracy_score
from sklearn.preprocessing import StandardScaler

#1. Load the dataset
#Example dataset: UCI ML Spam dataset (simulated as CSV or
you can download it)
#For now, let's assume you have 'spam.csv' with the last
column named 'spam' (1=spam, 0= not spam)
  df = pd.read_csv('spam.csv')

# Preview the dataset
print("Dataset Preview:")
print(df.head())

#2. Prepare features and labels
X= df.drop('spam', axis=1) #Features (drop the target column)
Y= df['spam'] # Target

#3. Split the data
X_train, X_test, Y_train, y_test = train_test_split(X,y,
test_size =0.3, random_state=42)
```

```
#4 Feature Scaling (important for Logistic Regression)
Scaler = StandardScaler()
X_train_scaled = scaler.fit_transform (X_train)
X_test_scaled = Scaler.transform (X_test)

#5 Train Logistic Regression model
model = Logistic Regression ()
model.fit (X_train_scaled, y_train)

#6 Predictions
y_pred = model.predict (X_test_scaled)

#7. Evaluation
print ("\n Confusion Matrix :")
print (Confusion_matrix (y_test, y_pred))
print ("\n Accuracy Score :")
print (accuracy_score (y_test, y_pred))
```

[illegible]

4th program

```
Dataset Preview:
   word_freq_make  word_freq_address  word_freq_all  word_freq_3d  \
0            0.00               0.64           0.64           0.0
1            0.21               0.28           0.50           0.0
2            0.06               0.00           0.71           0.0
3            0.00               0.00           0.00           0.0
4            0.00               0.00           0.00           0.0

   word_freq_our  word_freq_over  word_freq_remove  word_freq_internet  \
0           0.32            0.00              0.00                0.00
1           0.14            0.28              0.21                0.07
2           1.23            0.19              0.19                0.12
3           0.63            0.00              0.31                0.63
4           0.63            0.00              0.31                0.63

   word_freq_order  word_freq_mail  ...  char_freq_;  char_freq_(  \
0             0.00            0.00  ...         0.00        0.000
1             0.00            0.94  ...         0.00        0.132
2             0.64            0.25  ...         0.01        0.143
3             0.31            0.63  ...         0.00        0.137
4             0.31            0.63  ...         0.00        0.135

   char_freq_[  char_freq_!  char_freq_$  char_freq_#  \
0          0.0        0.778        0.000        0.000
1          0.0        0.372        0.180        0.048
2          0.0        0.276        0.184        0.010
3          0.0        0.137        0.000        0.000
```

[illegible]

```
4       0.0     0.135     0.000     0.000

   capital_run_length_average  capital_run_length_longest  \
0                       3.756                          61
1                       5.114                         101
2                       9.821                         485
3                       3.537                          40
4                       3.537                          40

   capital_run_length_total  spam
0                       278     1
1                      1028     1
2                      2259     1
3                       191     1
4                       191     1

[5 rows x 58 columns]

Confusion Matrix:
[[769  35]
 [ 71 506]]

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.96      0.94       804
           1       0.94      0.88      0.91       577

    accuracy                           0.92      1381
   macro avg       0.93      0.92      0.92      1381
weighted avg       0.92      0.92      0.92      1381


Accuracy Score:
0.9232440260680667
```

[illegible]